

# Foundation of Cryptography, Lecture 7

## Non-Interactive ZK and Proof of Knowledge

Iftach Haitner, Tel Aviv University

Tel Aviv University.

May 28, 2013

# Part I

## Non-Interactive Zero Knowledge

## Interaction is crucial for $\mathcal{ZK}$

### Claim 1

Assume that  $\mathcal{L} \subseteq \{0, 1\}^*$  has a **one-message  $\mathcal{ZK}$**  proof (even computational), with standard completeness and soundness,<sup>a</sup> then  $\mathcal{L} \in \mathcal{BPP}$ .

---

<sup>a</sup>That is, the completeness is  $\frac{2}{3}$  and soundness error is  $\frac{1}{3}$ .

# Interaction is crucial for $\mathcal{ZK}$

## Claim 1

Assume that  $\mathcal{L} \subseteq \{0, 1\}^*$  has a **one-message  $\mathcal{ZK}$**  proof (even computational), with standard completeness and soundness,<sup>a</sup> then  $\mathcal{L} \in \mathcal{BPP}$ .

---

<sup>a</sup>That is, the completeness is  $\frac{2}{3}$  and soundness error is  $\frac{1}{3}$ .

Proof: HW

# Interaction is crucial for $\mathcal{ZK}$

## Claim 1

Assume that  $\mathcal{L} \subseteq \{0, 1\}^*$  has a **one-message  $\mathcal{ZK}$**  proof (even computational), with standard completeness and soundness,<sup>a</sup> then  $\mathcal{L} \in \mathcal{BPP}$ .

---

<sup>a</sup>That is, the completeness is  $\frac{2}{3}$  and soundness error is  $\frac{1}{3}$ .

Proof: HW

- 1 To reduce interaction we **relax** the zero-knowledge requirement

# Interaction is crucial for $\mathcal{ZK}$

## Claim 1

Assume that  $\mathcal{L} \subseteq \{0, 1\}^*$  has a **one-message  $\mathcal{ZK}$**  proof (even computational), with standard completeness and soundness,<sup>a</sup> then  $\mathcal{L} \in \mathcal{BPP}$ .

---

<sup>a</sup>That is, the completeness is  $\frac{2}{3}$  and soundness error is  $\frac{1}{3}$ .

Proof: HW

① To reduce interaction we **relax** the zero-knowledge requirement

① Witness Indistinguishability

$\{\langle (P(w_x^1), V^*)(x) \rangle\}_{x \in \mathcal{L}} \approx_c \{\langle (P(w_x^2), V^*)(x) \rangle\}_{x \in \mathcal{L}},$   
for any  $\{w_x^1 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$  and  $\{w_x^2 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

# Interaction is crucial for $\mathcal{ZK}$

## Claim 1

Assume that  $\mathcal{L} \subseteq \{0, 1\}^*$  has a **one-message  $\mathcal{ZK}$**  proof (even computational), with standard completeness and soundness,<sup>a</sup> then  $\mathcal{L} \in \mathcal{BPP}$ .

---

<sup>a</sup>That is, the completeness is  $\frac{2}{3}$  and soundness error is  $\frac{1}{3}$ .

Proof: HW

- ➊ To reduce interaction we **relax** the zero-knowledge requirement
  - ➊ Witness Indistinguishability
$$\{\langle (P(w_x^1), V^*)(x) \rangle\}_{x \in \mathcal{L}} \approx_c \{\langle (P(w_x^2), V^*)(x) \rangle\}_{x \in \mathcal{L}},$$
for any  $\{w_x^1 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$  and  $\{w_x^2 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$
  - ➋ Witness Hiding

# Interaction is crucial for $\mathcal{ZK}$

## Claim 1

Assume that  $\mathcal{L} \subseteq \{0, 1\}^*$  has a **one-message  $\mathcal{ZK}$**  proof (even computational), with standard completeness and soundness,<sup>a</sup> then  $\mathcal{L} \in \mathcal{BPP}$ .

---

<sup>a</sup>That is, the completeness is  $\frac{2}{3}$  and soundness error is  $\frac{1}{3}$ .

Proof: HW

- ➊ To reduce interaction we **relax** the zero-knowledge requirement
  - ➊ Witness Indistinguishability
$$\{\langle (P(w_x^1), V^*)(x) \rangle\}_{x \in \mathcal{L}} \approx_c \{\langle (P(w_x^2), V^*)(x) \rangle\}_{x \in \mathcal{L}},$$
for any  $\{w_x^1 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$  and  $\{w_x^2 \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$
  - ➋ Witness Hiding
  - ➌ Non-interactive “zero knowledge”



# Non-Interactive Zero Knowledge ( $\mathcal{NIZK}$ )

## Definition 2 ( $\mathcal{NIZK}$ )

A pair of **non interactive** PPTM's  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L} \in \mathcal{NP}$ , if  $\exists \ell \in \text{poly}$  s.t.

- **Completeness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P(x, w(x), c)) = 1] \geq 2/3$ ,  
where  $w(x) \in R_{\mathcal{L}}(x)$  for any  $x \in \mathcal{L}$  ( $w$  is an arbitrary function)
- **Soundness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$ ,  
for any  $P^*$  and  $x \notin \mathcal{L}$
- $\mathcal{ZK}$ :  $\exists$  PPTM  $S$  s.t.  
 $\{(x, c, P(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$

# Non-Interactive Zero Knowledge ( $\mathcal{NIZK}$ )

## Definition 2 ( $\mathcal{NIZK}$ )

A pair of **non interactive** PPTM's  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L} \in \mathcal{NP}$ , if  $\exists \ell \in \text{poly}$  s.t.

- **Completeness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P(x, w(x), c)) = 1] \geq 2/3$ ,  
where  $w(x) \in R_{\mathcal{L}}(x)$  for any  $x \in \mathcal{L}$  ( $w$  is an arbitrary function)
- **Soundness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$ ,  
for any  $P^*$  and  $x \notin \mathcal{L}$
- $\mathcal{ZK}$ :  $\exists$  PPTM  $S$  s.t.  
 $\{(x, c, P(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$

- $c$  – common (random) reference string (CRS)

# Non-Interactive Zero Knowledge ( $\mathcal{NIZK}$ )

## Definition 2 ( $\mathcal{NIZK}$ )

A pair of **non interactive** PPTM's  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L} \in \mathcal{NP}$ , if  $\exists \ell \in \text{poly}$  s.t.

- **Completeness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P(x, w(x), c)) = 1] \geq 2/3$ ,  
where  $w(x) \in R_{\mathcal{L}}(x)$  for any  $x \in \mathcal{L}$  ( $w$  is an arbitrary function)
- **Soundness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$ ,  
for any  $P^*$  and  $x \notin \mathcal{L}$
- $\mathcal{ZK}$ :  $\exists$  PPTM  $S$  s.t.  
 $\{(x, c, P(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$

- $c$  – common (random) reference string (CRS)
- CRS is chosen **by the simulator**

# Non-Interactive Zero Knowledge ( $\mathcal{NIZK}$ )

## Definition 2 ( $\mathcal{NIZK}$ )

A pair of **non interactive** PPTM's  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L} \in \mathcal{NP}$ , if  $\exists \ell \in \text{poly}$  s.t.

- **Completeness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P(x, w(x), c)) = 1] \geq 2/3$ ,  
where  $w(x) \in R_{\mathcal{L}}(x)$  for any  $x \in \mathcal{L}$  ( $w$  is an arbitrary function)
- **Soundness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$ ,  
for any  $P^*$  and  $x \notin \mathcal{L}$
- $\mathcal{ZK}$ :  $\exists$  PPTM  $S$  s.t.  
 $\{(x, c, P(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$

- $c$  – common (random) reference string (CRS)
- CRS is chosen **by the simulator**
- What does the definition stand for?

# Non-Interactive Zero Knowledge ( $\mathcal{NIZK}$ )

## Definition 2 ( $\mathcal{NIZK}$ )

A pair of **non interactive** PPTM's  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L} \in \mathcal{NP}$ , if  $\exists \ell \in \text{poly}$  s.t.

- **Completeness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P(x, w(x), c)) = 1] \geq 2/3$ ,  
where  $w(x) \in R_{\mathcal{L}}(x)$  for any  $x \in \mathcal{L}$  ( $w$  is an arbitrary function)
- **Soundness:**  $\Pr_{c \leftarrow \{0,1\}^{\ell(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$ ,  
for any  $P^*$  and  $x \notin \mathcal{L}$
- $\mathcal{ZK}$ :  $\exists$  PPTM  $S$  s.t.  
 $\{(x, c, P(x, w(x), c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{\ell(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$

- $c$  – common (random) reference string (CRS)
- CRS is chosen **by the simulator**
- What does the definition stand for?
- Amplification?

# Section 1

## **NIZK in HBM**

## Hidden Bits Model (HBM)

A CRS is chosen at random, but **only the prover can see it**. The prover chooses which bits to reveal as part of the proof.

## Hidden Bits Model (HBM)

A CRS is chosen at random, but **only the prover can see it**. The prover chooses which bits to reveal as part of the proof.

Let  $c^H$  be the “hidden” CRS:

- Prover sees  $c^H$ , and outputs a proof  $\pi$  and a set of indices  $\mathcal{I}$
- Verifier only sees the bits in  $c^H$  that are indexed by  $\mathcal{I}$
- Simulator outputs a proof  $\pi$ , a set of indices  $\mathcal{I}$  and a partially hidden CRS  $c^H$



## Hidden Bits Model (HBM)

A CRS is chosen at random, but **only the prover can see it**. The prover chooses which bits to reveal as part of the proof.

Let  $c^H$  be the “hidden” CRS:

- Prover sees  $c^H$ , and outputs a proof  $\pi$  and a set of indices  $\mathcal{I}$
- Verifier only sees the bits in  $c^H$  that are indexed by  $\mathcal{I}$
- Simulator outputs a proof  $\pi$ , a set of indices  $\mathcal{I}$  and a partially hidden CRS  $c^H$
- Soundness, completeness and ZK are naturally defined.

## Hidden Bits Model (HBM)

A CRS is chosen at random, but **only the prover can see it**. The prover chooses which bits to reveal as part of the proof.

Let  $c^H$  be the “hidden” CRS:

- Prover sees  $c^H$ , and outputs a proof  $\pi$  and a set of indices  $\mathcal{I}$
- Verifier only sees the bits in  $c^H$  that are indexed by  $\mathcal{I}$
- Simulator outputs a proof  $\pi$ , a set of indices  $\mathcal{I}$  and a partially hidden CRS  $c^H$
- Soundness, completeness and ZK are naturally defined.
- We give a  $\mathcal{NIZK}$  for  $\mathcal{HC}$ , Directed Graph Hamiltonicity, in the **HBM**, and then transfer it into a  $\mathcal{NIZK}$  for  $\mathcal{HC}$  in the **standard model**.

## Hidden Bits Model (HBM)

A CRS is chosen at random, but **only the prover can see it**. The prover chooses which bits to reveal as part of the proof.

Let  $c^H$  be the “hidden” CRS:

- Prover sees  $c^H$ , and outputs a proof  $\pi$  and a set of indices  $\mathcal{I}$
- Verifier only sees the bits in  $c^H$  that are indexed by  $\mathcal{I}$
- Simulator outputs a proof  $\pi$ , a set of indices  $\mathcal{I}$  and a partially hidden CRS  $c^H$
- Soundness, completeness and ZK are naturally defined.
- We give a  $\mathcal{NIZK}$  for  $\mathcal{HC}$ , Directed Graph Hamiltonicity, in the **HBM**, and then transfer it into a  $\mathcal{NIZK}$  for  $\mathcal{HC}$  in the **standard model**.
- The latter (standard model)  $\mathcal{NIZK}$  for  $\mathcal{HC}$  implies a  $\mathcal{NIZK}$  for all  $\mathcal{NP}$

## Useful Matrix

- **Permutation matrix:** an  $n \times n$  Boolean matrix, where each row/column contains a single 1

## Useful Matrix

- **Permutation matrix**: an  $n \times n$  Boolean matrix, where each row/column contains a single 1
- **Hamiltonian matrix**: an  $n \times n$  adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)

## Useful Matrix

- **Permutation matrix**: an  $n \times n$  Boolean matrix, where each row/column contains a single 1
- **Hamiltonian matrix**: an  $n \times n$  adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An  $n^3 \times n^3$  Boolean matrix is **useful**: if it contains  $n \times n$  Hamiltonian *generalized* sub-matrix, and all its other entries are zeros

# Useful Matrix

- **Permutation matrix**: an  $n \times n$  Boolean matrix, where each row/column contains a single 1
- **Hamiltonian matrix**: an  $n \times n$  adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An  $n^3 \times n^3$  Boolean matrix is **useful**: if it contains  $n \times n$  Hamiltonian *generalized* sub-matrix, and all its other entries are zeros

## Useful Matrix

- **Permutation matrix**: an  $n \times n$  Boolean matrix, where each row/column contains a single 1
- **Hamiltonian matrix**: an  $n \times n$  adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An  $n^3 \times n^3$  Boolean matrix is **useful**: if it contains  $n \times n$  Hamiltonian *generalized* sub-matrix, and all its other entries are zeros

### Claim 3

Let  $T$  be a random  $n^3 \times n^3$  Boolean matrix where each entry is 1 w.p  $n^{-5}$ . Then,  $\Pr[T \text{ is useful}] \in \Omega(n^{-3/2})$ .



## Proving Claim 3

- The expected # of ones (entries) in  $T$  is  $n^6 \cdot n^{-5} = n$ .

## Proving Claim 3

- The expected # of ones (entries) in  $T$  is  $n^6 \cdot n^{-5} = n$ .
- By extended Chernoff bound,  $T$  contains exactly  $n$  ones w.p.  $\theta(1/\sqrt{n})$ .

## Proving Claim 3

- The expected # of ones (entries) in  $T$  is  $n^6 \cdot n^{-5} = n$ .
- By extended Chernoff bound,  $T$  contains exactly  $n$  ones w.p.  $\theta(1/\sqrt{n})$ .
- Each row/column of  $T$  contain more than a single one entry with probability at most  $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$ .  
Hence, wp at least  $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$ , no row or column of  $T$  contains more than a single one entry.

## Proving Claim 3

- The expected # of ones (entries) in  $T$  is  $n^6 \cdot n^{-5} = n$ .
- By extended Chernoff bound,  $T$  contains exactly  $n$  ones w.p.  $\theta(1/\sqrt{n})$ .
- Each row/column of  $T$  contain more than a single one entry with probability at most  $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$ .  
Hence, wp at least  $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$ , no row or column of  $T$  contains more than a single one entry.
- Hence, wp  $\theta(1/\sqrt{n})$  the matrix  $T$  contains a permutation matrix and all its other entries are zero.

## Proving Claim 3

- The expected # of ones (entries) in  $T$  is  $n^6 \cdot n^{-5} = n$ .
- By extended Chernoff bound,  $T$  contains exactly  $n$  ones w.p.  $\theta(1/\sqrt{n})$ .
- Each row/column of  $T$  contain more than a single one entry with probability at most  $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$ .  
Hence, wp at least  $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$ , no row or column of  $T$  contains more than a single one entry.
- Hence, wp  $\theta(1/\sqrt{n})$  the matrix  $T$  contains a permutation matrix and all its other entries are zero.
- A random permutation matrix forms a cycle wp  $1/n$  (there are  $n!$  permutation matrices and  $(n-1)!$  of them form a cycle)

## $\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph  $G = ([n], E)$

## $\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph  $G = ([n], E)$
- we assume wlg. that  $n$  is a power of 2

## $\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph  $G = ([n], E)$
- we assume wlg. that  $n$  is a power of 2
- Common reference string  $T$  viewed as a  $n^3 \times n^3$  Boolean matrix, where each entry is 1 w.p  $n^{-5}$  ??



## $\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph  $G = ([n], E)$
- we assume wlg. that  $n$  is a power of 2
- Common reference string  $T$  viewed as a  $n^3 \times n^3$  Boolean matrix, where each entry is 1 w.p  $n^{-5}$  ??

## $\mathcal{NIZK}$ for Hamiltonicity in HBM

- Common input: a directed graph  $G = ([n], E)$
- we assume w.l.g. that  $n$  is a power of 2
- Common reference string  $T$  viewed as a  $n^3 \times n^3$  Boolean matrix, where each entry is 1 w.p  $n^{-5}$  ??

### Algorithm 4 (P)

Input:  $G$  and a cycle  $C$  in  $G$ . A CRS  $T \in \{0, 1\}_{n^3 \times n^3}$

- 1 If  $T$  not useful, set  $\mathcal{I} = n^3 \times n^3$  (i.e., reveal all  $T$ ) and  $\phi = \perp$   
Otherwise, let  $H$  be the (generalized)  $n \times n$  sub-matrix containing the hamiltonian cycle in  $T$ .
- 2 Set  $\mathcal{I} = T \setminus H$  (i.e., reveal the bits of  $T$  outside of  $H$ )
- 3 Choose  $\phi \leftarrow \Pi_n$  s.t.  $C$  is mapped to the cycle in  $H$
- 4 Add all the entries in  $H$  corresponding to non edges in  $G$  (with respect to  $\phi$ ) to  $\mathcal{I}$
- 5 Output  $\pi = (\mathcal{I}, \phi)$

### Algorithm 5 (V)

Input: a graph  $G$ , index set  $\mathcal{I} \subseteq [n^3] \times [n^3]$ , ordered set  $\{T_i\}_{i \in \mathcal{I}}$  and a mapping  $\phi$

- 1 **Accept** if all the bits of  $T$  are revealed and  $T$  is **not useful**.  
Otherwise,
- 2 Verify that  $\exists n \times n$  submatrix  $H \subseteq T$  with all entries in  $T \setminus H$  are zeros.
- 3 Verify that  $\phi \in \Pi_n$ , and that all entries of  $H$  **not corresponding to edges of  $G$**  (according to  $\phi$ ) are zeros

## $\mathcal{NIZK}$ for Hamiltonicity in HBM cont.

### Algorithm 5 (V)

Input: a graph  $G$ , index set  $\mathcal{I} \subseteq [n^3] \times [n^3]$ , ordered set  $\{T_i\}_{i \in \mathcal{I}}$  and a mapping  $\phi$

- 1 **Accept** if all the bits of  $T$  are revealed and  $T$  is **not useful**.  
Otherwise,
- 2 Verify that  $\exists n \times n$  submatrix  $H \subseteq T$  with all entries in  $T \setminus H$  are zeros.
- 3 Verify that  $\phi \in \Pi_n$ , and that all entries of  $H$  **not corresponding to edges of  $G$**  (according to  $\phi$ ) are zeros

### Claim 6

The above protocol is a **perfect**  $\mathcal{NIZK}$  for  $\mathcal{HC}$  in the HBM, with **perfect** completeness and soundness error  $1 - \Omega(n^{-3/2})$

## Proving Claim 6

- Completeness: Clear

## Proving Claim 6

- Completeness: Clear
- Soundness: Assume  $T$  is useful and  $V$  accepts. Then  $\phi^{-1}$  maps the unrevealed “edges” of  $H$  to the edges of  $G$ .  
Hence,  $\phi^{-1}$  maps the cycle in  $H$  to an Hamiltonian cycle in  $G$

## Proving Claim 6

- Completeness: Clear
- Soundness: Assume  $T$  is useful and  $V$  accepts. Then  $\phi^{-1}$  maps the unrevealed “edges” of  $H$  to the edges of  $G$ .  
Hence,  $\phi^{-1}$  maps the cycle in  $H$  to an Hamiltonian cycle in  $G$
- Zero knowledge?

## Algorithm 7 (S)

Input:  $G$

- 1 Choose  $T$  at random (i.e., each entry is one wp  $n^{-5}$ ).
- 2 If  $T$  is not useful, set  $\mathcal{I} = n^3 \times n^3$  and  $\phi = \perp$ .  
Otherwise,
  - 3 Set  $\mathcal{I} = T \setminus H$  (where  $H$  is the hamiltonian sub-matrix in  $T$ )
  - 4 Let  $\phi \leftarrow \Pi_n$ . Replace **all** entries of  $H$  with zeros
  - 5 Add the entries in  $H$  corresponding to **non edges** in  $G$  to  $\mathcal{I}$
  - 6 Output  $\pi = (T, \mathcal{I}, \phi)$



## Algorithm 7 (S)

Input:  $G$

- 1 Choose  $T$  at random (i.e., each entry is one wp  $n^{-5}$ ).
- 2 If  $T$  is not useful, set  $\mathcal{I} = n^3 \times n^3$  and  $\phi = \perp$ .  
Otherwise,
- 3 Set  $\mathcal{I} = T \setminus H$  (where  $H$  is the hamiltonian sub-matrix in  $T$ )
- 4 Let  $\phi \leftarrow \Pi_n$ . Replace **all** entries of  $H$  with zeros
- 5 Add the entries in  $H$  corresponding to **non edges** in  $G$  to  $\mathcal{I}$
- 6 Output  $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful  $T$ 's.

## Algorithm 7 (S)

Input:  $G$

- 1 Choose  $T$  at random (i.e., each entry is one wp  $n^{-5}$ ).
- 2 If  $T$  is not useful, set  $\mathcal{I} = n^3 \times n^3$  and  $\phi = \perp$ .  
Otherwise,
- 3 Set  $\mathcal{I} = T \setminus H$  (where  $H$  is the hamiltonian sub-matrix in  $T$ )
- 4 Let  $\phi \leftarrow \Pi_n$ . Replace all entries of  $H$  with zeros
- 5 Add the entries in  $H$  corresponding to non edges in  $G$  to  $\mathcal{I}$
- 6 Output  $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful  $T$ 's.
- For useful  $T$ , the location of  $H$  is uniform in the real and simulated case.

## Algorithm 7 (S)

Input:  $G$

- 1 Choose  $T$  at random (i.e., each entry is one wp  $n^{-5}$ ).
- 2 If  $T$  is not useful, set  $\mathcal{I} = n^3 \times n^3$  and  $\phi = \perp$ .  
Otherwise,
- 3 Set  $\mathcal{I} = T \setminus H$  (where  $H$  is the hamiltonian sub-matrix in  $T$ )
- 4 Let  $\phi \leftarrow \Pi_n$ . Replace all entries of  $H$  with zeros
- 5 Add the entries in  $H$  corresponding to non edges in  $G$  to  $\mathcal{I}$
- 6 Output  $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful  $T$ 's.
- For useful  $T$ , the location of  $H$  is uniform in the real and simulated case.
- $\phi$  is a random element in  $\Pi_n$  in both (real and simulated) cases

## Algorithm 7 (S)

Input:  $G$

- 1 Choose  $T$  at random (i.e., each entry is one wp  $n^{-5}$ ).
- 2 If  $T$  is not useful, set  $\mathcal{I} = n^3 \times n^3$  and  $\phi = \perp$ .  
Otherwise,
- 3 Set  $\mathcal{I} = T \setminus H$  (where  $H$  is the hamiltonian sub-matrix in  $T$ )
- 4 Let  $\phi \leftarrow \Pi_n$ . Replace all entries of  $H$  with zeros
- 5 Add the entries in  $H$  corresponding to non edges in  $G$  to  $\mathcal{I}$
- 6 Output  $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful  $T$ 's.
- For useful  $T$ , the location of  $H$  is uniform in the real and simulated case.
- $\phi$  is a random element in  $\Pi_n$  in both (real and simulated) cases
- Hence, the simulation is perfect!

## Section 2

# **From HBM to Standard NIZK**

# Trapdoor Permutations

## Definition 8 (trapdoor permutations)

A triplet  $(G, f, \text{Inv})$ , where  $G$  is a PPTM, and  $f$  and  $\text{Inv}$  are polynomial-time computable functions, is a **family of trapdoor permutation (TDP)**, if:

- 1 On input  $1^n$ ,  $G(1^n)$  outputs a pair  $(sk, pk)$ .
- 2  $f_{pk} = f(pk, \cdot)$  is a permutation over  $\{0, 1\}^n$ , for every  $n \in \mathbb{N}$  and  $pk \in \text{Supp}(G(1^n)_2)$ .
- 3  $\text{Inv}(sk, \cdot) \equiv f_{pk}^{-1}$  for every  $(sk, pk) \in \text{Supp}(G(1^n))$
- 4 For any PPTM  $A$ ,  $\Pr_{x \leftarrow \{0, 1\}^n, pk \leftarrow G(1^n)_2} [A(pk, x) = f_{pk}^{-1}(x)] = \text{neg}(n)$

# Hardcore Predicates for Trapdoor Permutations

## Definition 9 (hardcore predicates for TDP)

A polynomial-time computable  $b: \{0, 1\}^n \mapsto \{0, 1\}$  is a **hardcore predicate** of a TDP  $(G, f, \text{Inv})$ , if

$$\Pr_{e \leftarrow G(1^n)_2, x \leftarrow \{0, 1\}^n} [P(e, f_e(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPTM  $P$ .

# Hardcore Predicates for Trapdoor Permutations

## Definition 9 (hardcore predicates for TDP)

A polynomial-time computable  $b: \{0, 1\}^n \mapsto \{0, 1\}$  is a **hardcore predicate** of a TDP  $(G, f, \text{Inv})$ , if

$$\Pr_{e \leftarrow G(1^n)_2, x \leftarrow \{0, 1\}^n} [P(e, f_e(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPTM  $P$ .

Goldreich-Levin: any TDP has an hardcore predicate (ignoring padding issues)



## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$  (equals  $(p-1)(q-1)$  for  $n = pq$  with  $p, q \in \mathcal{P}$ )

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$  (equals  $(p-1)(q-1)$  for  $n = pq$  with  $p, q \in \mathcal{P}$ )
- For every  $e \in \mathbb{Z}_{\phi(n)}^*$ , the function  $f(x) \equiv x^e \bmod n$  is a permutation over  $\mathbb{Z}_n^*$ .

In particular,  $(x^e)^d \equiv x \bmod n$ , for every  $x \in \mathbb{Z}_n^*$ , where  $d \equiv e^{-1} \bmod \phi(n)$

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$  (equals  $(p-1)(q-1)$  for  $n = pq$  with  $p, q \in \mathcal{P}$ )
- For every  $e \in \mathbb{Z}_{\phi(n)}^*$ , the function  $f(x) \equiv x^e \bmod n$  is a permutation over  $\mathbb{Z}_n^*$ .

In particular,  $(x^e)^d \equiv x \bmod n$ , for every  $x \in \mathbb{Z}_n^*$ , where  $d \equiv e^{-1} \bmod \phi(n)$

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$  (equals  $(p-1)(q-1)$  for  $n = pq$  with  $p, q \in \mathcal{P}$ )
- For every  $e \in \mathbb{Z}_{\phi(n)}^*$ , the function  $f(x) \equiv x^e \bmod n$  is a permutation over  $\mathbb{Z}_n^*$ .

In particular,  $(x^e)^d \equiv x \bmod n$ , for every  $x \in \mathbb{Z}_n^*$ , where  $d \equiv e^{-1} \bmod \phi(n)$

### Definition 10 (RSA)

- $G(p, q)$  sets  $pk = (n = pq, e)$  for some  $e \in \mathbb{Z}_{\phi(n)}^*$ , and  $sk = (n, d \equiv e^{-1} \bmod \phi(n))$
- $f(pk, x) = x^e \bmod n$
- $\text{Inv}(sk, x) = x^d \bmod n$

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$  (equals  $(p-1)(q-1)$  for  $n = pq$  with  $p, q \in \mathcal{P}$ )
- For every  $e \in \mathbb{Z}_{\phi(n)}^*$ , the function  $f(x) \equiv x^e \bmod n$  is a permutation over  $\mathbb{Z}_n^*$ .

In particular,  $(x^e)^d \equiv x \bmod n$ , for every  $x \in \mathbb{Z}_n^*$ , where  $d \equiv e^{-1} \bmod \phi(n)$

### Definition 10 (RSA)

- $G(p, q)$  sets  $pk = (n = pq, e)$  for some  $e \in \mathbb{Z}_{\phi(n)}^*$ , and  $sk = (n, d \equiv e^{-1} \bmod \phi(n))$
- $f(pk, x) = x^e \bmod n$
- $\text{Inv}(sk, x) = x^d \bmod n$

Factoring is easy  $\implies$  RSA is easy.

## Example, RSA

In the following  $n \in \mathbb{N}$  and all operations are modulo  $n$ .

- $\mathbb{Z}_n = [n]$  and  $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$  (equals  $(p-1)(q-1)$  for  $n = pq$  with  $p, q \in \mathcal{P}$ )
- For every  $e \in \mathbb{Z}_{\phi(n)}^*$ , the function  $f(x) \equiv x^e \bmod n$  is a permutation over  $\mathbb{Z}_n^*$ .

In particular,  $(x^e)^d \equiv x \bmod n$ , for every  $x \in \mathbb{Z}_n^*$ , where  $d \equiv e^{-1} \bmod \phi(n)$

### Definition 10 (RSA)

- $G(p, q)$  sets  $pk = (n = pq, e)$  for some  $e \in \mathbb{Z}_{\phi(n)}^*$ , and  $sk = (n, d \equiv e^{-1} \bmod \phi(n))$
- $f(pk, x) = x^e \bmod n$
- $\text{Inv}(sk, x) = x^d \bmod n$

Factoring is easy  $\implies$  RSA is easy. Other direction?



## The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .

## The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .
- Let  $(G, f, \text{Inv})$  be a TDP and let  $b$  be an hardcore bit for it. For simplicity we assume  $G(1^n)$  chooses  $(sk, pk)$  as follows

# The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .
- Let  $(G, f, \text{Inv})$  be a TDP and let  $b$  be an hardcore bit for it. For simplicity we assume  $G(1^n)$  chooses  $(sk, pk)$  as follows
  - 1  $sk \leftarrow \{0, 1\}^n$

# The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .
- Let  $(G, f, \text{Inv})$  be a TDP and let  $b$  be an hardcore bit for it. For simplicity we assume  $G(1^n)$  chooses  $(sk, pk)$  as follows
  - 1  $sk \leftarrow \{0, 1\}^n$
  - 2  $pk = PK(sk)$

# The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .
- Let  $(G, f, \text{Inv})$  be a TDP and let  $b$  be an hardcore bit for it. For simplicity we assume  $G(1^n)$  chooses  $(sk, pk)$  as follows
  - 1  $sk \leftarrow \{0, 1\}^n$
  - 2  $pk = PK(sk)$

# The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .
- Let  $(G, f, \text{Inv})$  be a TDP and let  $b$  be an hardcore bit for it. For simplicity we assume  $G(1^n)$  chooses  $(sk, pk)$  as follows
  - 1  $sk \leftarrow \{0, 1\}^n$
  - 2  $pk = PK(sk)$

where  $PK: \{0, 1\}^n \mapsto \{0, 1\}^n$  is a polynomial-time computable function.

# The transformation

- Let  $(P_H, V_H)$  be a HBM  $\mathcal{NIZK}$  for  $\mathcal{L}$ , and let  $\ell(n)$  be the length of the CRS used for  $x \in \{0, 1\}^n$ .
- Let  $(G, f, \text{Inv})$  be a TDP and let  $b$  be an hardcore bit for it. For simplicity we assume  $G(1^n)$  chooses  $(sk, pk)$  as follows
  - 1  $sk \leftarrow \{0, 1\}^n$
  - 2  $pk = PK(sk)$

where  $PK: \{0, 1\}^n \mapsto \{0, 1\}^n$  is a polynomial-time computable function.

We construct a  $\mathcal{NIZK}(P, V)$  for  $\mathcal{L}$ , with the same completeness and “not too large” soundness error.

# The protocol

## Algorithm 11 (P)

Input:  $x \in \mathcal{L}$ ,  $w \in R_{\mathcal{L}}(x)$  and CRS  $c = (c_1, \dots, c_{\ell}) \in \{0, 1\}^{n\ell}$ , where  $n = |x|$  and  $\ell = \ell(n)$ .

- 1 Choose  $(sk, pk) \leftarrow G(sk)$  and compute  $c^H = (b(z_1 = f_{pk}^{-1}(c_1)), \dots, b(z_{\ell(n)} = f_{pk}^{-1}(c_{\ell})))$
- 2 Let  $(\pi_H, \mathcal{I}) \leftarrow P_H(x, w, c^H)$  and output  $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$



# The protocol

## Algorithm 11 (P)

Input:  $x \in \mathcal{L}$ ,  $w \in R_{\mathcal{L}}(x)$  and CRS  $c = (c_1, \dots, c_\ell) \in \{0, 1\}^{n\ell}$ , where  $n = |x|$  and  $\ell = \ell(n)$ .

- 1 Choose  $(sk, pk) \leftarrow G(sk)$  and compute  $c^H = (b(z_1 = f_{pk}^{-1}(c_1)), \dots, b(z_{\ell(n)} = f_{pk}^{-1}(c_\ell)))$
- 2 Let  $(\pi_H, \mathcal{I}) \leftarrow P_H(x, w, c^H)$  and output  $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$

## Algorithm 12 (V)

Input:  $x \in \mathcal{L}$ , CRS  $c = (c_1, \dots, c_\ell) \in \{0, 1\}^{np}$ , and  $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$ , where  $n = |x|$  and  $\ell = \ell(n)$ .

- 1 Verify that  $pk \in \{0, 1\}^n$  and that  $f_{pk}(z_i) = c_i$  for every  $i \in \mathcal{I}$
- 2 Return  $V_H(x, \pi_H, \mathcal{I}, c^H)$ , where  $c_i^H = b(z_i)$  for every  $i \in \mathcal{I}$ .

### Claim 13

Assuming that  $(P_H, V_H)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  in the HBM with soundness error  $2^{-n} \cdot \alpha$ , then  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  with the same completeness, and soundness error  $\alpha$ .

### Claim 13

Assuming that  $(P_H, V_H)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  in the HBM with soundness error  $2^{-n} \cdot \alpha$ , then  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  with the same completeness, and soundness error  $\alpha$ .

Proof: Assume for simplicity that  $b$  is unbiased (i.e.,  $\Pr[b(U_n) = 1] = \frac{1}{2}$ ).

### Claim 13

Assuming that  $(P_H, V_H)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  in the HBM with soundness error  $2^{-n} \cdot \alpha$ , then  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  with the same completeness, and soundness error  $\alpha$ .

Proof: Assume for simplicity that  $b$  is unbiased (i.e.,  $\Pr[b(U_n) = 1] = \frac{1}{2}$ ). For every  $pk \in \{0, 1\}^n$ :  $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0, 1\}^{np}}$  is uniformly distributed in  $\{0, 1\}^\ell$ .

### Claim 13

Assuming that  $(P_H, V_H)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  in the HBM with soundness error  $2^{-n} \cdot \alpha$ , then  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  with the same completeness, and soundness error  $\alpha$ .

Proof: Assume for simplicity that  $b$  is unbiased (i.e.,  $\Pr[b(U_n) = 1] = \frac{1}{2}$ ). For every  $pk \in \{0, 1\}^n$ :  $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0, 1\}^{np}}$  is uniformly distributed in  $\{0, 1\}^\ell$ .

- Completeness: clear

### Claim 13

Assuming that  $(P_H, V_H)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  in the HBM with soundness error  $2^{-n} \cdot \alpha$ , then  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  with the same completeness, and soundness error  $\alpha$ .

Proof: Assume for simplicity that  $b$  is unbiased (i.e.,  $\Pr[b(U_n) = 1] = \frac{1}{2}$ ). For every  $pk \in \{0, 1\}^n$ :  $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0, 1\}^{np}}$  is uniformly distributed in  $\{0, 1\}^\ell$ .

- Completeness: clear
- Soundness: follows by a union bound over all possible choice of  $pk \in \{0, 1\}^n$ .

### Claim 13

Assuming that  $(P_H, V_H)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  in the HBM with soundness error  $2^{-n} \cdot \alpha$ , then  $(P, V)$  is a  $\mathcal{NIZK}$  for  $\mathcal{L}$  with the same completeness, and soundness error  $\alpha$ .

Proof: Assume for simplicity that  $b$  is unbiased (i.e.,  $\Pr[b(U_n) = 1] = \frac{1}{2}$ ). For every  $pk \in \{0, 1\}^n$ :  $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_\ell))\right)_{c \leftarrow \{0, 1\}^{np}}$  is uniformly distributed in  $\{0, 1\}^\ell$ .

- Completeness: clear
- Soundness: follows by a union bound over all possible choice of  $pk \in \{0, 1\}^n$ .
- Zero knowledge:?

# Proving zero knowledge

## Algorithm 14 (S)

Input:  $x \in \{0, 1\}^n$  of length  $n$ .

- Let  $(\pi_H, \mathcal{I}, c^H) = S_H(x)$ , where  $S_H$  is the simulator of  $(P_H, V_H)$
- Output  $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$ , where
  - ▶  $pk \leftarrow G(U_n)$
  - ▶ Each  $z_i$  is chosen at random in  $\{0, 1\}^n$  such that  $b(z_i) = c_i^H$
  - ▶  $c_i = f_{pk}(z_i)$  for  $i \in \mathcal{I}$ , and a random value in  $\{0, 1\}^n$  otherwise.



# Proving zero knowledge

## Algorithm 14 (S)

Input:  $x \in \{0, 1\}^n$  of length  $n$ .

- Let  $(\pi_H, \mathcal{I}, c^H) = S_H(x)$ , where  $S_H$  is the simulator of  $(P_H, V_H)$
- Output  $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$ , where
  - ▶  $pk \leftarrow G(U_n)$
  - ▶ Each  $z_i$  is chosen at random in  $\{0, 1\}^n$  such that  $b(z_i) = c_i^H$
  - ▶  $c_i = f_{pk}(z_i)$  for  $i \in \mathcal{I}$ , and a random value in  $\{0, 1\}^n$  otherwise.
- Exists efficient  $M$  s.t.  $M(S_H(x)) \equiv S(x)$  and  $M(P_H(x, w_x)) \approx_c P(x, w_x)$

# Proving zero knowledge

## Algorithm 14 (S)

Input:  $x \in \{0, 1\}^n$  of length  $n$ .

- Let  $(\pi_H, \mathcal{I}, c^H) = S_H(x)$ , where  $S_H$  is the simulator of  $(P_H, V_H)$
- Output  $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$ , where
  - ▶  $pk \leftarrow G(U_n)$
  - ▶ Each  $z_i$  is chosen at random in  $\{0, 1\}^n$  such that  $b(z_i) = c_i^H$
  - ▶  $c_i = f_{pk}(z_i)$  for  $i \in \mathcal{I}$ , and a random value in  $\{0, 1\}^n$  otherwise.
- Exists efficient  $M$  s.t.  $M(S_H(x)) \equiv S(x)$  and  $M(P_H(x, w_x)) \approx_c P(x, w_x)$
- Distinguishing  $P(x, w_x)$  from  $S(x)$  is hard

# Proving zero knowledge

## Algorithm 14 (S)

Input:  $x \in \{0, 1\}^n$  of length  $n$ .

- Let  $(\pi_H, \mathcal{I}, c^H) = S_H(x)$ , where  $S_H$  is the simulator of  $(P_H, V_H)$
- Output  $(c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$ , where
  - ▶  $pk \leftarrow G(U_n)$
  - ▶ Each  $z_i$  is chosen at random in  $\{0, 1\}^n$  such that  $b(z_i) = c_i^H$
  - ▶  $c_i = f_{pk}(z_i)$  for  $i \in \mathcal{I}$ , and a random value in  $\{0, 1\}^n$  otherwise.
- Exists efficient  $M$  s.t.  $M(S_H(x)) \equiv S(x)$  and  $M(P_H(x, w_x)) \approx_c P(x, w_x)$
- Distinguishing  $P(x, w_x)$  from  $S(x)$  is hard
- Need to be slightly modified to get “adaptive  $\mathcal{NIZK}$ ”

## Section 3

# **Adaptive NIZK**

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c))), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$



## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c))), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$ :  $\exists$  pair of PPTM's  $(S_1, S_2)$  s.t.  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{C} \cap \{0, 1\}^n$

$$\{(f(c), c, P(f(c), w_{f(c)}), c \leftarrow \{0, 1\}^{\ell(n)})\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.$$

where  $S^f(n)$  is the output of the following process

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$ :  $\exists$  pair of PPTM's  $(S_1, S_2)$  s.t.  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{C} \cap \{0, 1\}^n$

$$\{(f(c), c, P(f(c), w_{f(c)}), c \leftarrow \{0, 1\}^{\ell(n)})\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.$$

where  $S^f(n)$  is the output of the following process

①  $(c, s) \leftarrow S_1(1^n)$

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$ :  $\exists$  pair of PPTM's  $(S_1, S_2)$  s.t.  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{C} \cap \{0, 1\}^n$

$$\{(f(c), c, P(f(c), w_{f(c)}), c \leftarrow \{0, 1\}^{\ell(n)})\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.$$

where  $S^f(n)$  is the output of the following process

- 1  $(c, s) \leftarrow S_1(1^n)$
- 2  $x = f(c)$

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$ :  $\exists$  pair of PPTM's  $(S_1, S_2)$  s.t.  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{C} \cap \{0, 1\}^n$

$$\{(f(c), c, P(f(c), w_{f(c)}), c \leftarrow \{0, 1\}^{\ell(n)})\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.$$

where  $S^f(n)$  is the output of the following process

- 1  $(c, s) \leftarrow S_1(1^n)$
- 2  $x = f(c)$
- 3 Output  $(x, c, S_2(x, c, s))$

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$ :  $\exists$  pair of PPTM's  $(S_1, S_2)$  s.t.  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{C} \cap \{0, 1\}^n$

$$\{(f(c), c, P(f(c), w_{f(c)}), c \leftarrow \{0, 1\}^{\ell(n)})\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.$$

where  $S^f(n)$  is the output of the following process

- 1  $(c, s) \leftarrow S_1(1^n)$
- 2  $x = f(c)$
- 3 Output  $(x, c, S_2(x, c, s))$

## Adaptive $\mathcal{NIZK}$

$x$  is chosen **after** the CRS.

- **Completeness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{L} \cap \{0, 1\}^n$ :  
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P(f(c), w(f(c)), c)) = 1] \geq 2/3$
- **Soundness:**  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^n$  and  $P^*$   
 $\Pr_{c \leftarrow \{0, 1\}^{\ell(n)}} [V(f(c), c, P^*(c)) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$
- $\mathcal{ZK}$ :  $\exists$  pair of PPTM's  $(S_1, S_2)$  s.t.  $\forall f: \{0, 1\}^{\ell(n)} \mapsto \mathcal{C} \cap \{0, 1\}^n$

$$\{(f(c), c, P(f(c), w_{f(c)}), c \leftarrow \{0, 1\}^{\ell(n)})\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}.$$

where  $S^f(n)$  is the output of the following process

- 1  $(c, s) \leftarrow S_1(1^n)$
- 2  $x = f(c)$
- 3 Output  $(x, c, S_2(x, c, s))$

- Adaptive completeness and soundness are easy to achieve from any non-adaptive  $\mathcal{NIZK}$ .

- Adaptive completeness and soundness are easy to achieve from any non-adaptive  $\mathcal{NIZK}$ .
- Not every  $\mathcal{NIZK}$  is adaptive (but the above protocol is).



- Adaptive completeness and soundness are easy to achieve from any non-adaptive  $\mathcal{NIZK}$ .
- Not every  $\mathcal{NIZK}$  is adaptive (but the above protocol is).

- Adaptive completeness and soundness are easy to achieve from any non-adaptive  $\mathcal{NIZK}$ .
- Not every  $\mathcal{NIZK}$  is adaptive (but the above protocol is).

### Theorem 15

*Assume TDP exist, then every  $\mathcal{NP}$  language has an adaptive  $\mathcal{NIZK}$  with perfect completeness and negligible soundness error.*

- Adaptive completeness and soundness are easy to achieve from any non-adaptive  $\mathcal{NIZK}$ .
- Not every  $\mathcal{NIZK}$  is adaptive (but the above protocol is).

### Theorem 15

*Assume TDP exist, then every  $\mathcal{NP}$  language has an adaptive  $\mathcal{NIZK}$  with perfect completeness and negligible soundness error.*

In the following, when saying adaptive  $\mathcal{NIZK}$ , we mean negligible completeness and soundness error.

## Section 4

# Simulation Sound NIZK

# Simulation Soundness

A  $\mathcal{NIZK}$  system  $(P, V)$  for  $\mathcal{L}$  has (one-time) simulation soundness, if  $\exists$  a pair of PPTM's  $S = (S_1, S_2)$  satisfying the  $\mathcal{ZK}$  property of  $P$  with respect to  $\mathcal{L}$ , such that the following holds  $\forall$  pair of PPTM's  $(P_1^*, P_2^*)$ : let

## Experiment 16 ( $\text{Exp}_{V,S,P^*}^n$ )

- 1  $(c, s) \leftarrow S_1(1^n)$
- 2  $(x, p) \leftarrow P_1^*(1^n, c)$
- 3  $\pi \leftarrow S_2(x, c, s)$
- 4  $(x', \pi') \leftarrow P_2^*(p, \pi)$
- 5 Output  $(c, x, \pi, x', \pi')$

# Simulation Soundness

A  $\mathcal{NIZK}$  system  $(P, V)$  for  $\mathcal{L}$  has (one-time) simulation soundness, if  $\exists$  a pair of PPTM's  $S = (S_1, S_2)$  satisfying the  $\mathcal{ZK}$  property of  $P$  with respect to  $\mathcal{L}$ , such that the following holds  $\forall$  pair of PPTM's  $(P_1^*, P_2^*)$ : let

## Experiment 16 ( $\text{Exp}_{V,S,P^*}^n$ )

- 1  $(c, s) \leftarrow S_1(1^n)$
- 2  $(x, p) \leftarrow P_1^*(1^n, c)$
- 3  $\pi \leftarrow S_2(x, c, s)$
- 4  $(x', \pi') \leftarrow P_2^*(p, \pi)$
- 5 Output  $(c, x, \pi, x', \pi')$

We require  $\Pr[(c, x, \pi, x', \pi') \leftarrow \text{Exp}_{V,S,P^*}^n : x' \notin \mathcal{L} \wedge V(x', \pi', c) = 1 \wedge (x', \pi') \neq (x, \pi)] = \text{neg}(n)$ .

- Even for  $x \notin \mathcal{L}$ , hard to generate additional false proofs

- Even for  $x \notin \mathcal{L}$ , hard to generate additional false proofs
- Definition only considers efficient provers



- Even for  $x \notin \mathcal{L}$ , hard to generate additional false proofs
- Definition only considers efficient provers
- $(P, V)$  might be adaptive or non-adaptive

- Even for  $x \notin \mathcal{L}$ , hard to generate additional false proofs
- Definition only considers efficient provers
- $(P, V)$  might be adaptive or non-adaptive
- Adaptive  $\mathcal{NIZK}$  guarantees weak type of simulation soundness (hard to fake proofs for simulated CRS)

- Even for  $x \notin \mathcal{L}$ , hard to generate additional false proofs
- Definition only considers efficient provers
- $(P, V)$  might be adaptive or non-adaptive
- Adaptive  $\mathcal{NIZK}$  guarantees weak type of simulation soundness (hard to fake proofs for simulated CRS)
- Does the adaptive  $\mathcal{NIZK}$  we seen in class have simulation soundness?

## Construction

We present a simulation sound  $\mathcal{NIZK} (P, V)$  for  $\mathcal{L} \in \mathcal{NP}$

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- 1 Strong signature scheme  $(\text{Gen}, \text{Sign}, \text{Vrfy})$  (one time suffice)

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- 1 Strong signature scheme  $(\text{Gen}, \text{Sign}, \text{Vrfy})$  (one time suffice)
- 2 Non-interactive, perfectly-binding commitment  $\text{Com}$

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- 1 Strong signature scheme  $(\text{Gen}, \text{Sign}, \text{Vrfy})$  (one time suffice)
- 2 Non-interactive, perfectly-binding commitment  $\text{Com}$ 
  - Pseudorandom range: for some  $\ell \in \text{poly}$   
$$\{\text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)})\}_{w \in \{0, 1\}^*} \approx_c \{u \leftarrow \{0, 1\}^{\ell(|w|)}\}_{w \in \{0, 1\}^*}$$

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- ① Strong signature scheme  $(\text{Gen}, \text{Sign}, \text{Vrfy})$  (one time suffice)
- ② Non-interactive, perfectly-binding commitment  $\text{Com}$ 
  - ▶ Pseudorandom range: for some  $\ell \in \text{poly}$   
 $\{\text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)})\}_{w \in \{0, 1\}^*} \approx_c \{u \leftarrow \{0, 1\}^{\ell(|w|)}\}_{w \in \{0, 1\}^*}$
  - \* implied by OWP (or TDP)



## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- 1 Strong signature scheme ( $\text{Gen}, \text{Sign}, \text{Vrfy}$ ) (one time suffice)
- 2 Non-interactive, perfectly-binding commitment  $\text{Com}$ 
  - ▶ Pseudorandom range: for some  $\ell \in \text{poly}$   
 $\{\text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)})\}_{w \in \{0,1\}^*} \approx_c \{u \leftarrow \{0, 1\}^{\ell(|w|)}\}_{w \in \{0,1\}^*}$
  - \* implied by OWP (or TDP)
  - ▶ Negligible support: a random string is a valid commitment only with negligible probability.

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- 1 Strong signature scheme ( $\text{Gen}, \text{Sign}, \text{Vrfy}$ ) (one time suffice)
- 2 Non-interactive, perfectly-binding commitment  $\text{Com}$ 
  - ▶ Pseudorandom range: for some  $\ell \in \text{poly}$   
 $\{\text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)})\}_{w \in \{0, 1\}^*} \approx_c \{u \leftarrow \{0, 1\}^{\ell(|w|)}\}_{w \in \{0, 1\}^*}$
  - \* implied by OWP (or TDP)
  - ▶ Negligible support: a random string is a valid commitment only with negligible probability.
  - \* achieved from **any** commitment scheme by committing to the **same** value many times

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- ① Strong signature scheme  $(\text{Gen}, \text{Sign}, \text{Vrfy})$  (one time suffice)
- ② Non-interactive, perfectly-binding commitment  $\text{Com}$ 
  - ▶ Pseudorandom range: for some  $\ell \in \text{poly}$   
 $\{\text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)})\}_{w \in \{0, 1\}^*} \approx_c \{u \leftarrow \{0, 1\}^{\ell(|w|)}\}_{w \in \{0, 1\}^*}$
  - \* implied by OWP (or TDP)
  - ▶ Negligible support: a random string is a valid commitment only with negligible probability.
  - \* achieved from **any** commitment scheme by committing to the **same** value many times
- ③ Adaptive  $\mathcal{NIZK}(\mathcal{P}_A, \mathcal{V}_A)$  for  
 $\mathcal{L}_A := \{(x, \text{com}, w) : x \in \mathcal{L} \vee \exists r \in \{0, 1\}^* : \text{com} = \text{Com}(w, r)\} \in \mathcal{NP}$

## Construction

We present a simulation sound  $\mathcal{NIZK}(\mathcal{P}, \mathcal{V})$  for  $\mathcal{L} \in \mathcal{NP}$

### Ingredients:

- 1 Strong signature scheme ( $\text{Gen}, \text{Sign}, \text{Vrfy}$ ) (one time suffice)
- 2 Non-interactive, perfectly-binding commitment  $\text{Com}$ 
  - ▶ Pseudorandom range: for some  $\ell \in \text{poly}$   
 $\{\text{Com}(w, r \leftarrow \{0, 1\}^{\ell(|w|)})\}_{w \in \{0, 1\}^*} \approx_c \{u \leftarrow \{0, 1\}^{\ell(|w|)}\}_{w \in \{0, 1\}^*}$
  - \* implied by OWP (or TDP)
  - ▶ Negligible support: a random string is a valid commitment only with negligible probability.
  - \* achieved from **any** commitment scheme by committing to the **same** value many times
- 3 Adaptive  $\mathcal{NIZK}(\mathcal{P}_A, \mathcal{V}_A)$  for  
 $\mathcal{L}_A := \{(x, \text{com}, w) : x \in \mathcal{L} \vee \exists r \in \{0, 1\}^* : \text{com} = \text{Com}(w, r)\} \in \mathcal{NP}$ 
  - \* adaptive **WI** suffices

## Algorithm 17 (P)

**Input:**  $x \in \mathcal{L}$  and  $w \in R_{\mathcal{L}}(x)$ , and CRS  $c = (c_1, c_2)$

- 1  $(sk, vk) \leftarrow \text{Gen}(1^{|x|})$
- 2  $\pi_A \leftarrow P_A((x, c_1, vk), w, c_2)$
- 3  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 4 Output  $\pi = (vk, \pi_A, \sigma)$

## Algorithm 17 (P)

**Input:**  $x \in \mathcal{L}$  and  $w \in R_{\mathcal{L}}(x)$ , and CRS  $c = (c_1, c_2)$

- 1  $(sk, vk) \leftarrow \text{Gen}(1^{|x|})$
- 2  $\pi_A \leftarrow P_A((x, c_1, vk), w, c_2)$
- 3  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 4 Output  $\pi = (vk, \pi_A, \sigma)$

## Algorithm 18 (V)

**Input:**  $x \in \{0, 1\}^*$ ,  $\pi = (vk, \pi_A, \sigma)$  and a CRS  $c = (c_1, c_2)$

Verify that  $\text{Vrfy}_{vk}((x, \pi), \sigma) = 1$  and  $V_A((x, c_1, vk), c_2, \pi_A) = 1$

### Algorithm 17 (P)

**Input:**  $x \in \mathcal{L}$  and  $w \in R_{\mathcal{L}}(x)$ , and CRS  $c = (c_1, c_2)$

- 1  $(sk, vk) \leftarrow \text{Gen}(1^{|x|})$
- 2  $\pi_A \leftarrow P_A((x, c_1, vk), w, c_2)$
- 3  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 4 Output  $\pi = (vk, \pi_A, \sigma)$

### Algorithm 18 (V)

**Input:**  $x \in \{0, 1\}^*$ ,  $\pi = (vk, \pi_A, \sigma)$  and a CRS  $c = (c_1, c_2)$

Verify that  $\text{Vrfy}_{vk}((x, \pi), \sigma) = 1$  and  $V_A((x, c_1, vk), c_2, \pi_A) = 1$

### Claim 19

The proof system  $(P, V)$  is an adaptive  $\mathcal{NIZK}$  for  $\mathcal{L}$  with one-time simulation soundness.

- **Adaptive Completeness:** Clear



## Proving Claim 19

- **Adaptive Completeness:** Clear
- **Adaptive  $\mathcal{ZK}$ :**

## Proving Claim 19

- **Adaptive Completeness:** Clear
- **Adaptive  $\mathcal{ZK}$ :**
  - ▶  $S_1(1^n)$ :

## Proving Claim 19

- **Adaptive Completeness:** Clear
- **Adaptive  $\mathcal{ZK}$ :**
  - ▶  $S_1(1^n)$ :
    - 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .

## Proving Claim 19

- **Adaptive Completeness:** Clear
- **Adaptive  $\mathcal{ZK}$ :**
  - ▶  $S_1(1^n)$ :
    - 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
    - 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random

## Proving Claim 19

- **Adaptive Completeness:** Clear
- **Adaptive  $\mathcal{ZK}$ :**
  - ▶  $S_1(1^n)$ :
    - 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
    - 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random
  - ▶  $S_2(x, c, s = (z, sk, vk))$ :

## Proving Claim 19

- **Adaptive Completeness:** Clear
- **Adaptive  $\mathcal{ZK}$ :**
  - ▶  $S_1(1^n)$ :
    - 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
    - 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random
  - ▶  $S_2(x, c, s = (z, sk, vk))$ :
    - 1 let  $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$

# Proving Claim 19

- **Adaptive Completeness:** Clear

- **Adaptive  $\mathcal{ZK}$ :**

- ▶  $S_1(1^n)$ :

- 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
- 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random

- ▶  $S_2(x, c, s = (z, sk, vk))$ :

- 1 let  $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$
- 2  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$

## Proving Claim 19

- **Adaptive Completeness:** Clear

- **Adaptive  $\mathcal{ZK}$ :**

- ▶  $S_1(1^n)$ :

- 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
- 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random

- ▶  $S_2(x, c, s = (z, sk, vk))$ :

- 1 let  $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$
- 2  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 3 Output  $\pi = (vk, \pi_A, \sigma)$



# Proving Claim 19

- **Adaptive Completeness:** Clear

- **Adaptive  $\mathcal{ZK}$ :**

- ▶  $S_1(1^n)$ :

- 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
- 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random

- ▶  $S_2(x, c, s = (z, sk, vk))$ :

- 1 let  $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$
- 2  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 3 Output  $\pi = (vk, \pi_A, \sigma)$

# Proving Claim 19

- **Adaptive Completeness:** Clear

- **Adaptive  $\mathcal{ZK}$ :**

- ▶  $S_1(1^n)$ :

- 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
- 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random

- ▶  $S_2(x, c, s = (z, sk, vk))$ :

- 1 let  $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$
- 2  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 3 Output  $\pi = (vk, \pi_A, \sigma)$

Proof follows by the adaptive WI of  $(P_A, V_A)$  and the pseudorandomness of  $\text{Com}$

# Proving Claim 19

- **Adaptive Completeness:** Clear

- **Adaptive  $\mathcal{ZK}$ :**

- ▶  $S_1(1^n)$ :

- 1 Let  $(sk, vk) \leftarrow \text{Gen}(1^n)$ ,  $z \leftarrow \{0, 1\}^{\ell(n)}$  and  $c_1 = \text{Com}(vk, z)$ .
- 2 Output  $(c = (c_1, c_2), s = (z, sk, vk))$ , where  $c_2$  is chosen uniformly at random

- ▶  $S_2(x, c, s = (z, sk, vk))$ :

- 1 let  $\pi_A \leftarrow P_A((x, c_1, vk), z, c_2)$
- 2  $\sigma \leftarrow \text{Sign}_{sk}(x, \pi_A)$
- 3 Output  $\pi = (vk, \pi_A, \sigma)$

Proof follows by the adaptive WI of  $(P_A, V_A)$  and the pseudorandomness of  $\text{Com}$

- ▶ **Adaptive soundness:** Implicit in the proof of simulation soundness, given below

## Proving simulation soundness

Let  $P^* = (P_1^*, P_2^*)$  be a pair of PPTM's attacking the simulation soundness of  $(V, S)$  with respect to  $\mathcal{L}$ , and let  $c = (c_1, c_2)$ ,  $x$ ,  $\pi$ ,  $x'$  and  $\pi' = (vk', \pi'_A, \sigma')$  be the values generated by a random execution of  $\text{Exp}_{V,S,P^*}^n$ .

## Proving simulation soundness

Let  $P^* = (P_1^*, P_2^*)$  be a pair of PPTM's attacking the simulation soundness of  $(V, S)$  with respect to  $\mathcal{L}$ , and let  $c = (c_1, c_2)$ ,  $x$ ,  $\pi$ ,  $x'$  and  $\pi' = (vk', \pi'_A, \sigma')$  be the values generated by a random execution of  $\text{Exp}_{V,S,P^*}^n$ .

Assuming  $\text{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$ ,  $x' \notin \mathcal{L}$  and  $(x', \pi') \neq (x, \pi)$ , then with save but negligible probability:

- $vk'$  is not the verification key appeared in  $\pi$

## Proving simulation soundness

Let  $P^* = (P_1^*, P_2^*)$  be a pair of PPTM's attacking the simulation soundness of  $(V, S)$  with respect to  $\mathcal{L}$ , and let  $c = (c_1, c_2)$ ,  $x$ ,  $\pi$ ,  $x'$  and  $\pi' = (vk', \pi'_A, \sigma')$  be the values generated by a random execution of  $\text{Exp}_{V,S,P^*}^n$ .

Assuming  $\text{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$ ,  $x' \notin \mathcal{L}$  and  $(x', \pi') \neq (x, \pi)$ , then with save but negligible probability:

- $vk'$  is not the verification key appeared in  $\pi$   
 $\implies \nexists r \in \{0, 1\}^* \text{ s.t. } c_1 = \text{Com}(vk', r)$

## Proving simulation soundness

Let  $P^* = (P_1^*, P_2^*)$  be a pair of PPTM's attacking the simulation soundness of  $(V, S)$  with respect to  $\mathcal{L}$ , and let  $c = (c_1, c_2)$ ,  $x$ ,  $\pi$ ,  $x'$  and  $\pi' = (vk', \pi'_A, \sigma')$  be the values generated by a random execution of  $\text{Exp}_{V,S,P^*}^n$ .

Assuming  $\text{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$ ,  $x' \notin \mathcal{L}$  and  $(x', \pi') \neq (x, \pi)$ , then with save but negligible probability:

- $vk'$  is not the verification key appeared in  $\pi$ 
  - $\implies \nexists r \in \{0, 1\}^*$  s.t.  $c_1 = \text{Com}(vk', r)$
  - $\implies x'_A = (x', c_1, vk') \notin \mathcal{L}_A$

## Proving simulation soundness

Let  $P^* = (P_1^*, P_2^*)$  be a pair of PPTM's attacking the simulation soundness of  $(V, S)$  with respect to  $\mathcal{L}$ , and let  $c = (c_1, c_2)$ ,  $x$ ,  $\pi$ ,  $x'$  and  $\pi' = (vk', \pi'_A, \sigma')$  be the values generated by a random execution of  $\text{Exp}_{V,S,P^*}^n$ .

Assuming  $\text{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$ ,  $x' \notin \mathcal{L}$  and  $(x', \pi') \neq (x, \pi)$ , then with save but negligible probability:

- $vk'$  is not the verification key appeared in  $\pi$ 
  - $\implies \nexists r \in \{0, 1\}^*$  s.t.  $c_1 = \text{Com}(vk', r)$
  - $\implies x'_A = (x', c_1, vk') \notin \mathcal{L}_A$



## Proving simulation soundness

Let  $P^* = (P_1^*, P_2^*)$  be a pair of PPTM's attacking the simulation soundness of  $(V, S)$  with respect to  $\mathcal{L}$ , and let  $c = (c_1, c_2)$ ,  $x$ ,  $\pi$ ,  $x'$  and  $\pi' = (vk', \pi'_A, \sigma')$  be the values generated by a random execution of  $\text{Exp}_{V,S,P^*}^n$ .

Assuming  $\text{Vrfy}_{vk'}((x', \pi'_A), \sigma') = 1$ ,  $x' \notin \mathcal{L}$  and  $(x', \pi') \neq (x, \pi)$ , then with save but negligible probability:

- $vk'$  is not the verification key appeared in  $\pi$

$$\implies \nexists r \in \{0, 1\}^* \text{ s.t. } c_1 = \text{Com}(vk', r)$$

$$\implies x'_A = (x', c_1, vk') \notin \mathcal{L}_A$$

Since  $c_2$  was chosen at random by  $S_1$ , the adaptive soundness of  $(P_A, V_A)$  yields that  $\Pr[V_A(x'_A, c_2, \pi'_A) = 1] = \text{neg}(n)$ .

Adaptive soundness?

# Part II

## Proof of Knowledge

## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

### Definition 20 (knowledge extractor)

Let  $(P, V)$  be an interactive proof  $\mathcal{L} \in \mathcal{NP}$ . A probabilistic machine  $E$  is a **knowledge extractor** for  $(P, V)$  and  $R_{\mathcal{L}}$  with error  $\eta: \mathbb{N} \mapsto \mathbb{R}$ , if  $\exists t \in \text{poly}$  s.t.  $\forall x \in \mathcal{L}$  and deterministic algorithm  $P^*$ ,  $E^{P^*}(x)$  runs in expected time bounded by  $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$  and outputs  $w \in R_{\mathcal{L}}(x)$ , where  $\delta(x) = \Pr[(P^*, V)(x) = 1]$ .

$(P, V)$  is a proof of knowledge for  $\mathcal{L}$  with error  $\eta$ ,

## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

### Definition 20 (knowledge extractor)

Let  $(P, V)$  be an interactive proof  $\mathcal{L} \in \mathcal{NP}$ . A probabilistic machine  $E$  is a **knowledge extractor** for  $(P, V)$  and  $R_{\mathcal{L}}$  with error  $\eta: \mathbb{N} \mapsto \mathbb{R}$ , if  $\exists t \in \text{poly}$  s.t.  $\forall x \in \mathcal{L}$  and deterministic algorithm  $P^*$ ,  $E^{P^*}(x)$  runs in expected time bounded by  $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$  and outputs  $w \in R_{\mathcal{L}}(x)$ , where  $\delta(x) = \Pr[(P^*, V)(x) = 1]$ .

$(P, V)$  is a proof of knowledge for  $\mathcal{L}$  with error  $\eta$ ,

- A property of  $V$

## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

### Definition 20 (knowledge extractor)

Let  $(P, V)$  be an interactive proof  $\mathcal{L} \in \mathcal{NP}$ . A probabilistic machine  $E$  is a **knowledge extractor** for  $(P, V)$  and  $R_{\mathcal{L}}$  with error  $\eta: \mathbb{N} \mapsto \mathbb{R}$ , if  $\exists t \in \text{poly}$  s.t.  $\forall x \in \mathcal{L}$  and deterministic algorithm  $P^*$ ,  $E^{P^*}(x)$  runs in expected time bounded by  $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$  and outputs  $w \in R_{\mathcal{L}}(x)$ , where  $\delta(x) = \Pr[(P^*, V)(x) = 1]$ .

$(P, V)$  is a proof of knowledge for  $\mathcal{L}$  with error  $\eta$ ,

- A property of  $V$
- Why do we need it?

## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

### Definition 20 (knowledge extractor)

Let  $(P, V)$  be an interactive proof  $\mathcal{L} \in \mathcal{NP}$ . A probabilistic machine  $E$  is a **knowledge extractor** for  $(P, V)$  and  $R_{\mathcal{L}}$  with error  $\eta: \mathbb{N} \mapsto \mathbb{R}$ , if  $\exists t \in \text{poly}$  s.t.  $\forall x \in \mathcal{L}$  and deterministic algorithm  $P^*$ ,  $E^{P^*}(x)$  runs in expected time bounded by  $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$  and outputs  $w \in R_{\mathcal{L}}(x)$ , where  $\delta(x) = \Pr[(P^*, V)(x) = 1]$ .

$(P, V)$  is a proof of knowledge for  $\mathcal{L}$  with error  $\eta$ ,

- A property of  $V$
- Why do we need it?

## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

### Definition 20 (knowledge extractor)

Let  $(P, V)$  be an interactive proof  $\mathcal{L} \in \mathcal{NP}$ . A probabilistic machine  $E$  is a **knowledge extractor** for  $(P, V)$  and  $R_{\mathcal{L}}$  with error  $\eta: \mathbb{N} \mapsto \mathbb{R}$ , if  $\exists t \in \text{poly}$  s.t.  $\forall x \in \mathcal{L}$  and deterministic algorithm  $P^*$ ,  $E^{P^*}(x)$  runs in expected time bounded by  $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$  and outputs  $w \in R_{\mathcal{L}}(x)$ , where  $\delta(x) = \Pr[(P^*, V)(x) = 1]$ .

$(P, V)$  is a proof of knowledge for  $\mathcal{L}$  with error  $\eta$ ,

- A property of  $V$
- Why do we need it? Proving that you know the password



## Proof of Knowledge

The protocol  $(P, V)$  is a **proof of knowledge** for  $\mathcal{L} \in \mathcal{NP}$ , if  $P$  convinces  $V$  to accept  $x$ , only if it “knows”  $w \in R_{\mathcal{L}}(x)$ .

### Definition 20 (knowledge extractor)

Let  $(P, V)$  be an interactive proof  $\mathcal{L} \in \mathcal{NP}$ . A probabilistic machine  $E$  is a **knowledge extractor** for  $(P, V)$  and  $R_{\mathcal{L}}$  with error  $\eta: \mathbb{N} \mapsto \mathbb{R}$ , if  $\exists t \in \text{poly}$  s.t.  $\forall x \in \mathcal{L}$  and deterministic algorithm  $P^*$ ,  $E^{P^*}(x)$  runs in expected time bounded by  $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$  and outputs  $w \in R_{\mathcal{L}}(x)$ , where  $\delta(x) = \Pr[(P^*, V)(x) = 1]$ .

$(P, V)$  is a proof of knowledge for  $\mathcal{L}$  with error  $\eta$ ,

- A property of  $V$
- Why do we need it? Proving that you know the password
- Relation to  $\mathcal{ZK}$

## Examples

### Claim 21

The  $\mathcal{ZK}$  proof we've seen in class for  $\mathcal{GI}$ , has a knowledge extractor with error  $\frac{1}{2}$ .

## Examples

### Claim 21

The  $\mathcal{ZK}$  proof we've seen in class for  $\mathcal{GI}$ , has a knowledge extractor with error  $\frac{1}{2}$ .

Proof: ?

# Examples

## Claim 21

The  $\mathcal{ZK}$  proof we've seen in class for  $\mathcal{GI}$ , has a knowledge extractor with error  $\frac{1}{2}$ .

Proof: ?

## Claim 22

The  $\mathcal{ZK}$  proof we've seen in class for  $3\text{COL}$ , has a knowledge extractor with error  $\frac{1}{|E|}$ .

# Examples

## Claim 21

The  $\mathcal{ZK}$  proof we've seen in class for  $\mathcal{GI}$ , has a knowledge extractor with error  $\frac{1}{2}$ .

Proof: ?

## Claim 22

The  $\mathcal{ZK}$  proof we've seen in class for  $3\text{COL}$ , has a knowledge extractor with error  $\frac{1}{|E|}$ .

Proof: ?